

OemToChar

Carefully manage unit sizes and buffer bounds checking

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2007 Cigital, Inc.

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 6569 bytes

Attack Category	<ul style="list-style-type: none">• Malicious Input
Vulnerability Category	<ul style="list-style-type: none">• Buffer Overflow• Multibyte Character
Software Context	<ul style="list-style-type: none">• String Parsing
Location	
Description	<p>The OemToChar family of functions does not do bounds checking and is subject to wide-character vulnerabilities.</p> <p>The OemToChar routines translate OEM characters into standard ANSI characters. That is, for OEM char #252 (a superscript 'n' as in 2^n), the routine translates it to a normal 'n'. Note that this is not necessarily reversible.</p> <p>The usage behavior is very nontypical and the routines do NOT behave the same as each other.</p> <p>Replace OemToChar with OemToCharBuff. Use the 'length' field as the *number of characters to translate* (i.e., the length of the source string). The destination string must have at least enough space to store that string length plus the null character. Note: Unlike OemToChar, OemToCharBuff does NOT stop when it encounters the end of a null-terminated string. Furthermore, unless the length was provided properly, the routine will not copy the null byte and might not null terminate the string.</p> <p>With narrow (single-byte) characters, the routine allows you to specify the source and destination string addresses the same, allowing for in-place translation.</p> <p>The remaining "BUFF" functions have standard bounds checking issues if you pass in the wrong size:</p> <p>OemToCharBuff OemToCharBuffA OemToCharBuffW OemToAnsiBuff</p>

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	OemToAnsiBuffA OemToAnsiBuffW		
APIs	Function Name		Comments
	OemToChar		
	OemToCharA		
	OemToCharBuff		
	OemToCharBuffA		
	OemToCharBuffW		
	OemToCharW		
	OemToAnsi		
	OemToAnsiA		
	OemToAnsiBuff		
	OemToAnsiBuffA		
	OemToAnsiBuffW		
	OemToAnsiW		
	Method of Attack	<p>Be very wary using these functions. Their behavior is very nontypical.</p> <p>The OemToChar function has a variety of problems:</p> <ul style="list-style-type: none">* no bounds checking is done, so an attacker can trigger a buffer overflow* differences between single-byte and wide-character behavior can be misused <p>Also, for OemToCharBuff:</p> <ul style="list-style-type: none">* the wide-character versions accept a buffer length in <code>_characters,</code> not <code>_bytes.</code> If the user passes in the wrong size, the routine may think the buffer is twice its actual size and therefore overrun the buffer.	
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	Whenever there is a need to convert from OEM characters to ASCII.	Use OemToCharBuff instead of OemToChar. Ensure that all parameters are specified correctly.	Effective.
Signature Details	BOOL OemToChar(LPCSTR lpszSrc, LPTSTR lpszDst);		

Examples of Incorrect Code	<pre>TCHAR dst[15]; // Buffer is too small LPTSTR lpszDst = dst; if (! OemToChar(TEXT("String containing OEM characters"), lpszDst)) { /* handle error */ }</pre>	
Examples of Corrected Code	<pre>const TCHAR src[] = TEXT("String containing OEM characters"); LPCSTR lpszSrc = src; TCHAR dst[30]; LPTSTR lpszDst = dst; DWORD charsToConvert = strlen(lpszSrc) + 1; // should include terminating NULL if (charsToConvert > sizeof(dst)) { /* Handle error since buffer is not large enough */ } else { if (! OemToCharBuff(lpszSrc, lpszDst, charsToConvert)) { /* handle error */ } }</pre>	
Source References	<ul style="list-style-type: none">• Rough Auditing Tool for Security (RATS)²• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/winui/winui/windowsuserinterface/resources/strings/stringreference/stringfunctions/oemtochar.asp³	
Recommended Resources	<ul style="list-style-type: none">• MSDN reference for OemToChar⁴• MSDN reference for OemToCharBuff⁵	
Discriminant Set	Operating System	<ul style="list-style-type: none">• Windows
	Languages	<ul style="list-style-type: none">• C• C++

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

1. <mailto:copyright@cigital.com>

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.